

AUTOTUNING PARAMETER KENDALI PD DENGAN TSUKAMOTO FUZZY MENGUNAKAN BAHASA C

Muhammad Dedy Nurmansyah , Supriyono

Program Studi Elektronika Instrumentasi Jurusan Teknofisika Nuklir

Sekolah Tinggi Teknologi Nuklir – BATAN

Jl. Babarsari, Sleman, Yogyakarta

HP. 085643056315

Email: m.dedy.n@gmail.com

ABSTRAK

Kendali PD (*Proporsional-Differensial*) merupakan suatu metode pengendalian yang banyak digunakan di industri. Teknik ini memiliki dua parameter yaitu parameter kontrol proporsional (K_p) dan parameter kontrol differensial (K_d). Dalam metode konvensional, parameter PD di tuning dengan cara trial and error. Masalahnya, tuning parameter dengan cara trial and error tidak mudah dan membutuhkan waktu yang lama. Autotuning parameter dengan menggunakan logika fuzzy merupakan salah satu metode untuk memecahkan masalah ini. Metode ini memiliki dua masukan, yaitu error dan perubahan error. Penalaan konstanta PD dengan logika Fuzzy akan terus mengalami perubahan konstanta sesuai dengan perubahan nilai error dan delta error. Program Kendali Logika Fuzzy dibuat dalam bahasa C dan ditanamkan pada Mikrokontroler ATmega 16 sebagai kontroler dan diuji pada robot Line Follower (pengikut garis), bertujuan untuk mendapatkan respon sistem yang baik yaitu rise time yang cepat dan overshoot yang minimal, respon sistem stabil walaupun ditambahkan beberapa error ke dalam sistem, sistem selalu menuju error = 0 (ideal) artinya robot selalu berusaha mengikuti garis.

Kata Kunci : kendali PD, logika fuzzy, mikrokontroler Atmega16, bahasa C, robot line follower

1. PENDAHULUAN

1.1 Latar Belakang

Kontrol yang umum digunakan di dunia Industri adalah PID, PI dan PD *controler*. Kendali PD terdiri atas kontroler proporsional dan derivatif. Kendala yang sering dihadapi adalah bahwa sistem kendali konvensional ini membutuhkan pengetahuan tentang parameter-parameter sistem terlebih dahulu atau dapat dilakukan dengan cara coba-coba.

Permasalahan akan bertambah rumit jika digunakan untuk sistem yang kompleks. Kendali Logika Fuzzy dapat digunakan untuk membantu memudahkan penalaan parameter-parameter kendali konvensional tersebut.

Kendali PD ditala dengan menggunakan Kendali Logika Fuzzy model Tsukamoto. Program Kendali Logika Fuzzy dibuat dalam bahasa C menggunakan software *CodeVisionAVR*, program ditanamkan pada Mikrokontroler ATmega 16 sebagai kontroler dan diuji pada robot pengikut garis. Gabungan aksi kontrol proporsional dan aksi kontrol diferensial membentuk aksi kontrol proporsional-diferensial (PD *controller*).

Gabungan aksi ini mempunyai keunggulan dibandingkan dengan masing-masing kontrol yang menyusunnya. Keunggulan utamanya adalah diperolehnya keuntungan dari masing-masing aksi kontrol dan kekurangan dari aksi kontrol yang satu dapat diatasi dengan aksi kontrol yang lain. Elemen-elemen kontrol P bertujuan untuk mempercepat reaksi sebuah sistem (*rise time* yang cepat), sedangkan kontrol D bertujuan untuk mengurangi

overshoot akibat aksi kontrol P. Sehingga akan didapatkan respon sistem yang baik yaitu *rise time* yang cepat dan *overshoot* yang minimal.

1.2 Tujuan

Program *Autotuning* Kendali PD menggunakan Logika Fuzzy ditulis dalam bahasa C, kemudian ditanamkan dan diuji pada robot Fast Neutron, bertujuan untuk mendapatkan respon sistem yang baik yaitu *rise time* yang cepat dan *overshoot* yang minimal, dalam hal ini robot dapat mengikuti garis dengan stabil. Ketika robot menyimpang dari garis atau ada *error* pada sistem, kendali Logika Fuzzy melakukan *Autotuning* nilai parameter K_p dan K_d yang sesuai agar sistem selalu menuju *error* = 0 (ideal).

Program *Autotuning* Kendali PD menggunakan Logika Fuzzy yang ditulis dalam bahasa C ini diharapkan dapat diterapkan pada semua sistem kendali baik untuk robot/manipulator, maupun untuk mesin-mesin otomasi industri yang membutuhkan respon sistem yang stabil.

1.3 Metode

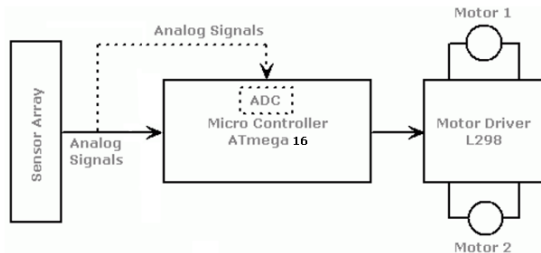
Metode penelitian yang dilakukan meliputi pengumpulan bahan dan peralatan yang dibutuhkan, mengumpulkan dan mempelajari literatur, mempelajari sistem dan prinsip kerja hardware, analisis dan perhitungan, membuat program dalam bahasa C dan menguji program pada hardware, menguji unjuk kerja sistem dan mengambil data

pengujian, membandingkan data pengujian dengan perhitungan, evaluasi dan perbaikan program.

2. TEORI

2.1 Robot Line Follower Fast Neutron

Salah satu aplikasi dari sistem kendali otomatis adalah robot *line follower*, yaitu robot yang dapat mengikuti garis. Pengembangan teknologi yang berbasis mikrokontroler dalam pembuatan robot ini dapat dijadikan dasar untuk membuat aplikasi yang lebih kompleks, seperti pengontrol pada mesin-mesin industri.



Gambar 1. Blok Diagram Hardware Elektronik Robot Fast Neutron

2.2 Kendali PD

Untuk mengimplementasikan sistem kendali PD pada mikrokontroler, PD harus diubah ke dalam persamaan diskrit dan diterjemahkan ke dalam bahasa program.

Formula untuk kontrol PD dalam program (Source Code C) adalah :

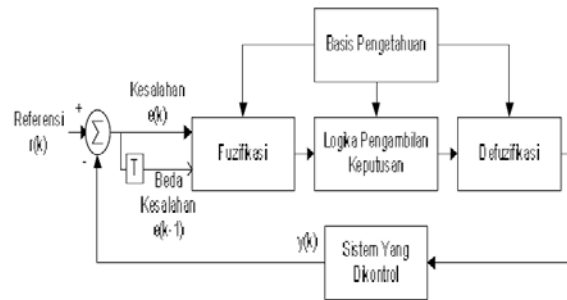
```
P = Kp * e;
d_e = e - last_e;
D = Kd * d_e;
last_e = e;
MV = P + D;
```

Dimana :

e = Nilai penyimpangan dari kondisi ideal (*Set Point – Sensor Measurement*)
 $last_e$ = Error sebelumnya
 d_e = Delta/Perubahan Error
 Kp = Parameter / Konstanta Proporsional
 Kd = Parameter / Konstanta Derivatif
 P = Galat P
 D = Galat D
 MV = Galat PD / output kendali PD (Nilai PWM motor untuk mengendalikan gerak Robot)

2.3 Logika Fuzzy

Kendali logika Fuzzy dilakukan dalam tiga tahap, yaitu fuzzifikasi, evaluasi aturan dan defuzzifikasi. Gambar 2 menunjukkan mekanisme kendali logika Fuzzy.



Gambar 2. Kendali logika Fuzzy kalang tertutup.

Komponen Fuzzifikasi berfungsi untuk memetakan masukan data tegas ke dalam himpunan Fuzzy menjadi nilai Fuzzy dari beberapa variabel linguistik masukan. Basis pengetahuan berisi pengetahuan sistem kendali sebagai pedoman evaluasi keadaan sistem untuk mendapatkan keluaran kendali sesuai yang diinginkan perancang. Basis pengetahuan terdiri dari basis data dan basis aturan Fuzzy. Basis data merupakan komponen untuk mendefinisikan himpunan Fuzzy dari masukan dan keluaran.

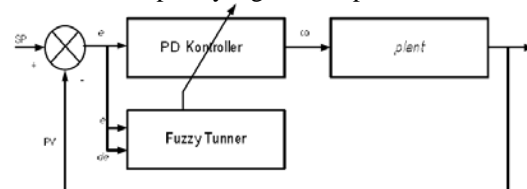
Basis Aturan Fuzzy merupakan kumpulan pernyataan aturan 'IF-THEN' yang didasarkan kepada pengetahuan pakar. Logika pengambilan keputusan disusun dengan cara menuliskan aturan yang menghubungkan antara masukan dan keluaran sistem Fuzzy. Aturan ini diekspresikan dalam kalimat: 'jika <masukan> maka <keluaran>'. Metode ini mempunyai bentuk aturan seperti persamaan berikut :

IF x is A and y is B then $z = k$
 Defuzzifikasi dapat didefinisikan sebagai proses pengubahan besaran Fuzzy yang disajikan dalam bentuk himpunan-himpunan Fuzzy keluaran dengan fungsi keanggotaan untuk mendapatkan kembali bentuk tegasnya (*crisp*).

3. PEMBAHASAN

3.1 Perancangan Sistem

Pada metode *Fuzzy Multiterm Controllers*, logika Fuzzy akan dipergunakan untuk menala atau mengadaptasi sebuah kontroler *multiterm* seperti kontroler PD seperti yang terlihat pada Gambar 3.

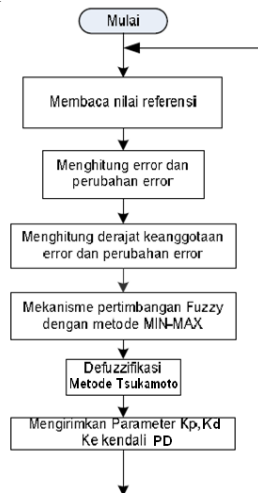


Gambar 3. Tuning Kontroler PD dengan fuzzy

Pendekatan dasar dalam mengadaptasi kontroler PD dengan menggunakan logika Fuzzy adalah dengan merencanakan *supervisory rules* dari Fuzzy tersebut dengan aturan (rule) berikut ini :

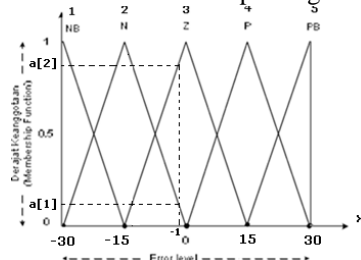
IF $e(k)$ is A_i and $de(k)$ is B_i , THEN Kp is C_i and Kd is D_i

Dimana, A_i , B_i , C_i , dan D_i adalah variabel linguistik dari *rule* ke- i ($i = 1, 2, \dots, m$) yang dapat direpresentasikan dengan fungsi keanggotaan tertentu. Program kendali Fuzzy terdiri atas Fuzzifikasi, evaluasi aturan, mekanisme pengambilan keputusan, dan defuzzifikasi. Keluaran pada proses defuzzifikasi merupakan hasil dari proses sistem Fuzzy secara keseluruhan. Blok diagram program Fuzzy sebagai *tuning* kendali PD diperlihatkan pada Gambar 4.

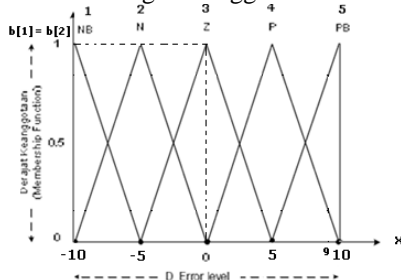


Gambar 4. Diagram blok utama sistem Fuzzy

Tahapan awal proses Fuzzifikasi adalah menentukan parameter-parameter fungsi keanggotaan pada setiap himpunan Fuzzy masukan. Pada pemrograman Fuzzifikasi ini digunakan parameter fungsi keanggotaan masukan berupa *error* dan *d_error*. Derajat keanggotaan bernilai antara 0 dan 1. Seluruh proses perancangan kendali logika Fuzzy dilakukan dengan menggunakan mikrokontroler ATmega 16 sehingga seluruh parameter fungsi keanggotaan *error* dan *d_error* sebagai input fuzzy pada Gambar 5 dan Gambar 6 dinormalisasi ke dalam bahasa pemrograman C.



Gambar 5. Fungsi keanggotaan *error* level



Gambar 6. Fungsi keanggotaan *d_error* level

Berdasarkan gambar 11 dan 12 diatas, input error dan *d_error* memiliki bentuk fungsi keanggotaan yang sama, yaitu sebagai berikut :

$$\begin{aligned} \mu_{NB} & \begin{cases} 1 & \longrightarrow x \leq NB \\ (N-x)/(N-NB) & \longrightarrow NB \leq x \leq N \\ 0 & \longrightarrow x \geq N \end{cases} \\ \mu_N & \begin{cases} 0 & \longrightarrow x \leq NB \text{ atau } x \geq 0 \\ (x-NB)/(N-NB) & \longrightarrow NB \leq x \leq N \\ (Z-x)/(Z-N) & \longrightarrow N \leq x \leq Z \end{cases} \\ \mu_Z & \begin{cases} 0 & \longrightarrow x \leq N \text{ atau } x \geq P \\ (x-N)/(Z-N) & \longrightarrow N \leq x \leq Z \\ (P-x)/(P-Z) & \longrightarrow Z \leq x \leq P \end{cases} \\ \mu_P & \begin{cases} 0 & \longrightarrow x \leq Z \text{ atau } x \geq PB \\ (x-Z)/(P-Z) & \longrightarrow Z \leq x \leq P \\ (PB-x)/(PB-P) & \longrightarrow P \leq x \leq PB \end{cases} \\ \mu_{PB} & \begin{cases} 0 & \longrightarrow x \leq P \\ (x-P)/(PB-P) & \longrightarrow P \leq x \leq PB \\ 1 & \longrightarrow x \geq PB \end{cases} \end{aligned}$$

Nilai parameter untuk input error dan *d_error* berbeda, dimana parameter input error yaitu e (sebagai nilai x), NB_e , N_e , Z_e , P_e dan PB_e sedangkan input *d_error* yaitu de (sebagai nilai x), NB_{de} , N_{de} , Z_{de} , P_{de} dan PB_{de} .

Nilai parameter ini disimpan di dalam memori eeprom mikrokontroler agar data tidak hilang walaupun catu daya mikrokontroler dimatikan. Berikut ini program inisialisasi parameter input error dan *d_error* dalam bahasa C (*Source Code C*).

```
//Inisialisasi Program Fuzzifikasi
//parameter error (input 1 fuzzy)
eeprom unsigned char NB_e = -30;
eeprom unsigned char N_e = -15;
eeprom unsigned char Z_e = -0;
eeprom unsigned char P_e = 15;
eeprom unsigned char PB_e = 30;

//parameter delta_error (input 2 fuzzy)
eeprom unsigned char NB_de = -10;
eeprom unsigned char N_de = -5;
eeprom unsigned char Z_de = -0;
eeprom unsigned char P_de = 5;
eeprom unsigned char PB_de = 10;
.....
```

Sedangkan program fungsi keanggotaan input error fuzzy dapat dibuat sebagai berikut:

```
//u_Negatif Besar
if (e<=NB_e)
{uNB_e=1;}
else if ((e>=NB_e)&&(e<=N_e))
{uNB_e=(N_e-e)/(N_e-NB_e);}
else if (e>=N_e)
{uNB_e=0;}
//u_Negatif
if ((e<=NB_e)|| (e>=Z_e))
{uN_e=0;}
else if ((e>=NB_e)&&(e<=N_e))
{uN_e=(e-NB_e)/(N_e-NB_e);}
else if ((e>=N_e)&&(e<=Z_e))
{uN_e=(Z_e-e)/(Z_e-N_e);}
.....
```

Program fungsi keanggotaan input *d_error* fuzzy sama seperti program fungsi keanggotaan input error tersebut, namun nilai parameter e , NB_e , N_e , Z_e , P_e dan PB_e harus diganti dengan de , NB_{de} , N_{de} , Z_{de} , P_{de} dan PB_{de} .

Setelah itu mengambil data $\mu(e)$ berdasarkan bentuk fungsi keanggotaan error diatas (gambar 11),

jumlah $\mu(e)$ maksimal yang dapat terbentuk pada gambar 11 tersebut adalah sebanyak dua buah $\mu(e)$, untuk memudahkan penulisan program, penulis memberi nama $a[1]$ dan $a[2]$, begitu pula dengan fungsi keanggotaan d_error (gambar 12) juga memiliki dua buah $\mu(de)$ yang diberi nama $b[1]$ dan $b[2]$. $\mu(e)$ dan $\mu(de)$ merupakan bilangan real (dalam bahasa C disebut *float*) yang bernilai 0 sampai dengan 1.

Selain mengambil data $\mu(e)$ dan $\mu(de)$, perlu dilakukan pula pengambilan data fuzzy pada input error dan d_error , bagaimana input error dan d_error dikatakan Negatif Besar (NB), Negatif (N), Zerro (Z), Positif (P) dan Positif Besar (PB), maka perlu diinisialisasikan.

Untuk memudahkan pemrograman, maka nilai fuzzy diinisialisasikan dalam bentuk angka dimana NB bernilai 1, N bernilai 2, Z bernilai 3, P bernilai 4, dan Z bernilai 5. Berdasarkan bentuk fungsi keanggotaan error (gambar 11), jumlah nilai fuzzy maksimal yang dapat terbentuk pada gambar 11 tersebut adalah sebanyak dua buah kemungkinan, penulis memberi nama $MFe[1]$ dan $MFe[2]$, sebagai contoh apabila $MFe[1]=1$ dan $MFe[2]=2$ artinya nilai fuzzy input error adalah Negatif Besar (NB) dan Negatif (N). Begitu pula dengan input d_error , nilai fuzzy untuk d_error penulis beri nama $MFde[1]$ dan $MFde[2]$. Penulisan programnya adalah sebagai berikut :

```
//Program Pengambilan data a[1], a[2] ,
//b[1], b[2], MFe[1], MFe[2], MFde[1] //dan
MFde[2].
unsigned char x=0;
float a[x]=0;
float b[x]=0;
unsigned char MFe[x]=0;
unsigned char MFde[x]=0;
float uNB_e,uN_e,uZ_e,uP_e,uPB_e,uNB_de
,uN_de,uZ_de,uP_de,uPB_de=0;
x=1;
if (uNB_e>0)
{a[x]=uNB_e;
MFe[x]=1;
x++;}
if (uN_e>0)
{a[x]=uN_e;
MFe[x]=2;
x++;}
if (uZ_e>0)
{a[x]=uZ_e;
MFe[x]=3;
x++;}
.....
x=0;
```

Metode pengambilan keputusan (inferensi) yang digunakan dalam pemrograman ini adalah metode Min-Max dengan mengambil nilai minimal antara fungsi keanggotaan input error dan d_error , yaitu : $\text{Min}(a[1],b[1])$, $\text{Min}(a[1],b[2])$, $\text{Min}(a[2],b[1])$ dan $\text{Min}(a[2],b[2])$. Hasil eksekusinya adalah $\alpha[1]$, $\alpha[2]$, $\alpha[3]$ dan $\alpha[4]$.

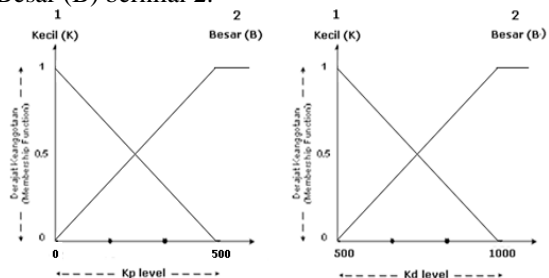
```
//Pengambilan Keputusan Metode
//Min-Max, alfa = Min(u e , u de)
float alfa[x]=1;
if (a[1]<b[1])
{alfa[1]=a[1];}
else
{alfa[1]=b[1];}

if (a[1]<b[2])
{alfa[2]=a[1];}
```

```
else
{alfa[2]=b[2];}
if (a[2]<b[1])
{alfa[3]=a[2];}
else
{alfa[3]=b[1];}
if (a[2]<b[2])
{alfa[4]=a[2];}
else
{alfa[4]=b[2];}
.....
```

Tahapan selanjutnya dalam *Fuzzy Inference System* adalah evaluasi aturan. Tujuan dari evaluasi aturan ini adalah menentukan derajat keanggotaan dari keluaran Fuzzy. Sebelum melakukan evaluasi aturan terlebih dahulu ditetapkan basis aturan. Basis aturan merupakan keseluruhan aturan dari kombinasi dua masukan yang mungkin. Secara lengkap, jumlah kombinasi yang mungkin dari dua himpunan Fuzzy masukan dengan masing-masing lima fungsi keanggotaan adalah dua puluh lima aturan. Basis aturan yang dibuat berdasarkan tingkah laku plant yang diinginkan. Keluaran Fuzzy akan menentukan nilai konstanta PD. Dua puluh lima aturan yang digunakan terlihat pada Tabel 1 dan Tabel 2.

Agar basis aturan dapat diterapkan dalam bentuk program, nilai fuzzy pada output juga diinisialisasikan dalam bentuk angka, dimana nilai fuzzy pada output yaitu Kecil (K) bernilai 1, dan Besar (B) bernilai 2.



Gambar 7. Fungsi keanggotaan Kp dan Kd level (Output Fuzzy)

Nilai fuzzy pada output Kp dan Kd di atas (gambar 13), penulis beri nama MF_{kp} dan MF_{kd} , sebagai contoh apabila $MF_{kp} = 1$ dan $MF_{kd} = 2$ artinya nilai Kp Kecil (K) dan Kd besar (B).

Tabel 1. Basis Aturan tuning PD dengan Logika Fuzzy untuk Konstanta Kp

Error	1 NB	2 N	3 Z	4 P	5 PB
1 NB	B	B	B	B	B
2 N	K	B	B	B	K
3 Z	K	K	K	K	K
4 P	K	B	B	B	K
5 PB	B	B	B	B	B

Tabel 2. Basis Aturan tuning PD dengan Logika Fuzzy untuk Konstanta Kd

Error	1	2	3	4	5
Error	NB	N	Z	P	PB
1	NB	B	B	B	B
2	N	K	B	B	K
3	Z	K	K	K	K
4	P	K	B	B	K
5	PB	B	B	B	B

Berdasarkan tabel diatas, maka program aturan fuzzy dalam bahasa C adalah sebagai berikut :

```
void aturan_fuzzy1()
{
//25 aturan fuzzy
if ((MFe[1]==1)&&(MFde[1]==1))
{MFkp=2;
MFkd=2;}
if ((MFe[1]==1)&&(MFde[1]==2))
{MFkp=2;
MFkd=2;}
if ((MFe[1]==1)&&(MFde[1]==3))
{MFkp=2;
MFkd=2;}
if ((MFe[1]==1)&&(MFde[1]==4))
{MFkp=2;
MFkd=2;}
if ((MFe[1]==1)&&(MFde[1]==5))
{MFkp=2;
MFkd=2;}
.....}
```

Karena memiliki dua buah $\mu(e)$, yaitu $a[1]$ dan $a[2]$, dan dua buah $\mu(de)$ yaitu $b[1]$ dan $b[2]$, maka diperlukan 4 subrutin aturan fuzzy seperti diatas, subrutin *void aturan_fuzzy1()* digunakan untuk aturan antara MFe[1] dan MFde[1], subrutin *void aturan_fuzzy2()* untuk aturan MFe[1] dan MFde[2], subrutin *void aturan_fuzzy3()* untuk aturan antara MFe[2] dan MFde[1], subrutin *void aturan_fuzzy4()* untuk aturan antara MFe[2] dan MFde [2]. Adapun mengenai isi aturan keempat subrutin tersebut sama seperti contoh subrutin *void aturan_fuzzy1()* di atas atau mengacu pada tabel 1 dan tabel 2 di atas.

Tahap terakhir dari inferensi Fuzzy adalah defuzzifikasi. Defuzzifikasi merupakan kebalikan dari proses Fuzzifikasi, yaitu mengubah himpunan Fuzzy keluaran menjadi keluaran tegas (*crisp*).

Pengubahan ini diperlukan karena konstanta kendali PD hanya mengenal nilai tegas sebagai variabel parameter. Nilai tegas (*crisp*) keluaran diperoleh dari himpunan-himpunan Fuzzy keluaran dengan menggunakan metode Tsukamoto.

Rumus Tsukamoto :

$$Output = \frac{alfa[1] * Z[1] + alfa[2] * Z[2] + alfa[3] * Z[3] + alfa[4] * Z[4]}{alfa[1] + alfa[2] + alfa[3] + alfa[4]}$$

Sebelum menghitung nilai tegas output Kp dan Kd, perlu dicari terlebih dahulu nilai Z[1], Z[2], Z[3], dan Z[4], yang didapatkan dari fungsi keanggotaan output Kp dan Kd level (Gambar 13), yaitu :

Output Kecil :

$$alfa[x] \begin{cases} 1 & \rightarrow Z[x] \leq K \\ (B-Z[x])/(B-K) & \rightarrow K \leq Z[x] \leq B \text{ sehingga } Z[x] = B - alfa[x]*(B-K) \\ 0 & \rightarrow Z[x] \geq B \end{cases}$$

Output Besar :

$$alfa[x] \begin{cases} 0 & \rightarrow Z[x] \leq K \\ (Z[x]-K)/(B-K) & \rightarrow K \leq Z[x] \leq B \text{ sehingga } Z[x] = alfa[x]*(B-K) + K \\ 1 & \rightarrow Z[x] \geq B \end{cases}$$

Untuk output Kp penulis beri nama kp_K untuk Kp yang bernilai Kecil dan kp_B untuk Kp yang bernilai Besar, sedangkan output Kd penulis beri nama kd_K untuk Kd yang bernilai Kecil dan kd_B untuk Kd yang bernilai Besar. Program Defuzzifikasi untuk output :

```
//Defuzzifikasi
unsigned char x=0;
float alfa[x];
float Z[x];
//parameter KP (output 1)
eeprom unsigned char kp_K = 0;
eeprom unsigned char kp_B = 500;
//parameter Kd (output 2)
eeprom unsigned char kd_K = 500;
eeprom unsigned char kd_B = 1000;
x=0;
while (x<5)
{x++;
switch (x)
{ case 1: aturan_fuzzy1();
break;
case 2: aturan_fuzzy2();
break;
case 3: aturan_fuzzy3();
break;
case 4: aturan_fuzzy4();
break; }
if (MFkp==1) // MFkp : kecil
{ if (alfa[x]==1)
{Z[x]=kp_K;}
else if (alfa[x]==0)
{Z[x]=kp_B;}
else
{Z[x]=kp_B-alfa[x]*(kp_B-kp_K);}}
else if (MFkp==2) // MFkp : besar
{ if (alfa[x]==0)
{Z[x]=kp_K;}
else if (alfa[x]==1)
{Z[x]=kp_B;}
else
{Z[x]=alfa[x]*(kp_B-kp_K)+kp_K;}}
Kp=(alfa[1]*Z[1]+alfa[2]*Z[2]+alfa[3]*Z[3]+alfa[4]*Z[4])/(alfa[1]+alfa[2]+alfa[3]+alfa[4]);
//output Kd
x=0;
.....}
```

3.2 Pengujian Sistem

Untuk menguji rancangan sistem *Autotuning* parameter kendali PD dengan Tsukamoto Fuzzy yang dibuat dalam bahasa C, dilakukan contoh perhitungan dari rancangan sistem yang telah dibuat, kemudian dibandingkan dengan hasil komputasi program yang ditanamkan pada hardware sebenarnya.

Pertanyaan 1 :

Apabila terjadi error = -1 dan $d_error = 0$ (lihat gambar 11 dan 12), berapakah nilai Kp dan Kd ?

Analisa Perhitungan :

Berdasarkan gambar 11, nilai fuzzy untuk error yaitu MFe[1] = Negatif (N) dan MFde[2] = Zerro (Z), sedangkan pada gambar 12, nilai d_error yaitu MFde[1] = Zerro (Z) dan MFde[2] = Zerro (Z). Berdasarkan aturan fuzzy pada tabel 1 dan 2 diperoleh data pada table 3.

Tabel 3. Aturan fuzzy untuk error=-1 dan $d_error=0$

Rule	error	d_error	Kp	Kd	
1	N(a[1])	Z(b[1])	Besar	Besar	alfa[1] ?
2	N(a[1])	b[2]=0			alfa[2]=0
3	Z(a[2])	Z(b[1])	Kecil	Kecil	alfa[3] ?
4	Z(a[2])	b[2]=0			alfa[4]=0

Rule 1

Berdasarkan fungsi keanggotaan error dan d_error pada gambar 11 dan 12, maka :

$$a[1] = \mu N = (Z-x)/(Z-N) = (0-(-1))/(0-(-15)) = 1/15 = 0.067$$

$$b[1] = \mu Z = (P-x)/(P-Z) = (15-0)/(15-0) = 15/15 = 1$$

$$alfa[1] = \min(\mu N, \mu Z) = \min(0.067, 1) = 0.067$$

Berdasarkan fungsi keanggotaan Kp dan Kd pada gambar 13, dimana Kp dan Kd Besar :

$$alfa[x] = (Z[x]-K)/(B-K)$$

$$Z[x] = alfa[x]*(B-K) + K, \text{ maka :}$$

Untuk Output Kp :

$$Z[1] = 0.067 * (500 - 0) + 0 = 33.33$$

Untuk Output Kd :

$$Z[1] = 0.067 * (1000-500) + 500 = 533.33$$

Rule 3

Berdasarkan fungsi keanggotaan error dan d_error pada gambar 11 dan 12, maka :

$$a[1] = \mu Z = (x-N)/(Z-N) = (-1 - (-15)) / (0 - (-15)) = 14 / 15 = 0.933$$

$$b[1] = \mu Z = (P-x)/(P-Z) = (15-0)/(15-0) = 15/15 = 1$$

$$alfa[3] = \min(\mu Z, \mu Z) = \min(0.933, 1) = 0.933$$

Berdasarkan fungsi keanggotaan Kp dan Kd pada gambar 13, dimana Kp dan Kd Kecil :

$$alfa[x] = (B-Z[x])/(B-K)$$

$$Z[x] = B - alfa[x]*(B-K), \text{ maka :}$$

Untuk Output Kp :

$$Z[3] = 500 - 0.933 * (500-0) = 33.5$$

Untuk Output Kd :

$$Z[3] = 1000 - 0.933 * (1000-500) = 533.5$$

$$Output = \frac{alfa[1] * Z[1] + alfa[2] * Z[2] + alfa[3] * Z[3] + alfa[4] * Z[4]}{alfa[1] + alfa[2] + alfa[3] + alfa[4]}$$

$$Kp = \frac{0.067 * 33.33 + 0 + 0.933 * 33.5 + 0}{0.067 + 0 + 0.933 + 0}$$

$$Kp = \frac{2.23311 + 0 + 31.2555 + 0}{1} = 33.48861$$

$$Kd = \frac{0.067 * 533.33 + 0 + 0.933 * 533.5 + 0}{0.067 + 0 + 0.933 + 0}$$

$$Kd = \frac{35.73311 + 0 + 497.7555 + 0}{1} = 533.48861$$

Pengujian program dilakukan dengan cara membaca dan menampilkan hasil komputasi program pada LCD display robot. Nilai Kp dan Kd pada hardware robot yang ditampilkan pada LCD setelah Program ditanamkan adalah sebagai berikut :



Gambar 8. Nilai Kp dan Kd untuk error=-1 dan d_error=0

Berdasarkan pengujian pada hardware, untuk error = -1 dan d_error = 0 maka Kp = 33.3 dan Kd = 533.3, nilai ini hampir sama dengan perhitungan, selisih disebabkan pembulatan yang dilakukan pada perhitungan.

4. KESIMPULAN

Dari hasil pengujian yang telah dilakukan, dapat diambil kesimpulan bahwa Program *Autotuning* Kendali PD menggunakan Logika Fuzzy metode Tsukamoto telah berhasil dibuat dalam bahasa C, dapat diterapkan pada semua sistem kendali baik untuk robot/manipulator, maupun untuk mesin-mesin otomatisasi industri yang membutuhkan respon sistem yang stabil yaitu *rise time* yang cepat dan *overshoot* yang minimal.

Dalam hal ini program ditanamkan dan diuji pada robot Line Follower Fast Neutron, memiliki respon sistem yang baik yaitu *rise time* yang cepat dan *overshoot* yang minimal, robot dapat mengikuti garis dengan stabil. Ketika robot menyimpang dari garis atau ada *error* pada sistem, kendali Logika Fuzzy melakukan *Autotuning* nilai parameter Kp dan Kd yang sesuai sehingga sistem selalu menuju *error* = 0 (ideal).

Kestabilan sistem ditentukan oleh aturan Fuzzy, bentuk dan nilai-nilai parameter fungsi keanggotaan input dan output Logika Fuzzy.

5. SARAN

Untuk mendapatkan kestabilan sistem yang optimal pada penerapan program *Autotuning* Kendali PD menggunakan Tsukamoto Fuzzy pada sistem yang berbeda, disarankan untuk mengubah nilai-nilai parameter fungsi keanggotaan input dan output Logika Fuzzy, atau mengubah bentuk dan menambah nilai parameter fungsi keanggotaan sampai mendapatkan respon sistem yang baik yaitu *rise time* yang cepat dan *overshoot* yang minimal.

PUSTAKA

- Wahyudi dkk (2008). Tuning Parameter Kontrol Proporsional-Integral Menggunakan Sugeno Fuzzy Inference System. *Jurnal Teknik Elektro, Jilid 10, Nomor 2*, 97-102.
- Iswari, L., dan Wahid, F. (2005). Alat Bantu Sistem Inferensi Fuzzy Metode Sugeno Orde Satu. *Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005)*.
- Budiharto, W. (2009). *Membuat Sendiri ROBOT CERDAS*. Jakarta: Elex Media Komputindo.
- Andrianto, H. (2008). *Pemrograman Mikrokontroler AVR ATMEGA 16 Menggunakan Bahasa C (CodeVision AVR)*. Bandung : Informatika.